Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	(("5925114") or ("6230119")).PN.	USPAT	OR	OFF	2004/11/22 13:54
L2	7	processor adj emulation adj system	USPAT	OR	OFF	2004/11/22 15:01
L3	155	((717/138) or (717/134) or (717/135)).CCLS.	USPAT	OR	OFF	2004/11/22 15:01
L4	13	3 and (target adj processor)	USPAT	OR	ON	2004/11/22 15:02

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	17	(instruction adj set adj simulator) and (emulator$4 or emulation$4) and (target$4 adj processor$4)	USPAT	OR	ON	2004/11/22 17:51

Terms used **emulation objects communications processor** Found **4** of **147,060**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 4 of 4

Relevance scale

1 The NuMesh: a modular, scalable communications substrate

Steve Ward, Karim Abdalla, Rajeev Dujari, Michael Fetterman, Frank Honoré, Ricardo Jenez, Philippe Laffont, Ken Mackenzie, Chris Metcalf, Milan Minsky, John Nguyen, John Pezaris, Gill Pratt, Russell Tessier

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Full text available: pdf(1.02 MB) Additional Information: full citation, references, index terms

2 The muse object architecture: a new operating system structuring concept

Yasuhiko Yokote, Fumio Teraoka, Atsushi Mitsuzawa, Nobuhisa Fujinami, Mario Tokoro

April 1991 **ACM SIGOPS Operating Systems Review**, Volume 25 Issue 2

Full text available: pdf(1.92 MB) Additional Information: full citation, abstract, citings, index terms

A next generation operating system should accommodate an ultra large-scale, open, self-advancing, and distributed environment. This environment is dynamic and versatile in nature. In it, an unlimited number of objects, ranging from fine to coarse-grained, are emerging, vanishing, evolving, and being replaced; computers of various processing capacities are dynamically connected and disconnected to networks; systems can optimize object execution by automatically detecting the user's and/or program ...

3 Web-based simulation: Web II: web-based simulation of systems described by partial differential equations

Manuel Alfonseca, Juan de Lara, Hans Vangheluwe

December 2001 **Proceedings of the 33nd conference on Winter simulation**

Full text available: pdf(538.55 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes how to take advantage of Internet services and object technology to solve 2D partial differential equations (PDEs) in a distributed manner. This is accomplished by means of a distributed object oriented continuous simulation language designed by our research group, called *OOCSMP,* and a Java (and C++) generating compiler for this language (called *C-OOL*). We also describe a graphical mesh generator, which produces *OOCSMP* code. The mesh generator may be inv ...

4 Automated synthesis of interface adapters for reusable classes

Satish R. Thatté

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

The need to fit together reusable components and system designs in spite of differences in protocol and representation choices occurs often in object-oriented software construction. It is therefore necessary to use adapters to achieve an exact fit between the available "socket" for a reusable part and the actual part. In this paper we discuss an approach to the

Terms used
processor emulator emulation software architecture different

Found **62,088** of **147,060**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Trace-driven memory simulation: a survey

Richard A. Uhlig, Trevor N. Mudge

June 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 2

Full text available: pdf(636.11 KB) Additional Information: full citation, abstract, references, citings, index terms, review

As the gap between processor and memory speeds continues to widen, methods for evaluating memory system designs before they are implemented in hardware are becoming increasingly important. One such method, trace-driven memory simulation, has been the subject of intense interest among researchers and has, as a result, enjoyed rapid development and substantial improvements during the past decade. This article surveys and analyzes these developments by establishing criteria for evaluating trac ...

Keywords: TLBs, caches, memory management, memory simulation, trace-driven simulation

2 The Microprogrammable Multi-Processor (MMP) system for simultaneous emulation of interoperating computer systems

Roy Mattson, Alan Salisbury

September 1974 **Conference record of the 7th annual workshop on Microprogramming**

Full text available: pdf(647.32 KB) Additional Information: full citation, abstract, references, citings, index terms

A Teleprocessing Design Center (TDC) has been established within the Communications/Automatic Laboratory of the US Army Electronics Command (ECOM) at Fort Monmouth, New Jersey, for the purpose of supporting experimentation in developing and validating Army Tactical Data Systems (ARTADS) configurations. The TDC is collocated within the Communications/Automatic Data Processing Laboratory with the Communications System Design Facility which includes circuit and message switching sys ...

3 Emulation - a useful tool in the development of computer systems

F. A. Salomon, D. A. Tafuri

March 1982 **Proceedings of the 15th annual symposium on Simulation**

Full text available: pdf(749.55 KB) Additional Information: full citation, abstract, references, index terms

Emulation is playing a key role in the development of a BELLMAC-32A microprocessor-based computer system at Bell Telephone Laboratories. The emulation is used for the development of the operating system for the new computer system to permit both hardware and software development to proceed in parallel. The emulation's goal is to permit the operating system to be developed before the hardware is available. This is aimed at reducing the time and effort required in the hardware/software integr ...

Terms used **emulation objects**

Found **12** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 12 of 12

Relevance scale

1 An interpretation of objects and object types

Martín Abadi, Luca Cardelli, Ramesh Viswanathan

January 1996 **Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

2 Automated synthesis of interface adapters for reusable classes

Satish R. Thatté

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

The need to fit together reusable components and system designs in spite of differences in protocol and representation choices occurs often in object-oriented software construction. It is therefore necessary to use adapters to achieve an exact fit between the available "socket" for a reusable part and the actual part. In this paper we discuss an approach to the construction of tools that largely automate the synthesis of adapter code. Such tools ...

3 Static program analysis: Improving program slicing with dynamic points-to data

Markus Mock, Darren C. Atkinson, Craig Chambers, Susan J. Eggers

November 2002 **Proceedings of the 10th ACM SIGSOFT symposium on Foundations of software engineering**

Full text available: pdf(109.02 KB) Additional Information: full citation, abstract, references, citings, index terms

Program slicing is a potentially useful analysis for aiding program understanding. However, slices of even small programs are often too large to be generally useful. Imprecise pointer analyses have been suggested as one cause of this problem. In this paper, we use dynamic points-to data, which represents optimal or optimistic pointer information, to obtain a bound on the best case slice size improvement that can be achieved with improved pointer precision. Our experiments show that slice size ca ...

Keywords: dynamic analysis, points-to analysis, program slicing

4 Session 4: static program analysis: Improving program slicing with dynamic points-to data

Markus Mock, Darren C. Atkinson, Craig Chambers, Susan J. Eggers

November 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 6

Full text available: pdf(1.05 MB) Additional Information: full citation, abstract, references, index terms

Program slicing is a potentially useful analysis for aiding program understanding. However,

slices of even small programs are often too large to be generally useful. Imprecise pointer analyses have been suggested as one cause of this problem. In this paper, we use dynamic points-to data, which represents optimal or optimistic pointer information, to obtain a bound on the best case slice size improvement that can be achieved with improved pointer precision. Our experiments show that slice size ca ...

Keywords: dynamic analysis, points-to analysis, program slicing

5 Web-based simulation: Web II: web-based simulation of systems described by partial differential equations

Manuel Alfonseca, Juan de Lara, Hans Vangheluwe
December 2001 **Proceedings of the 33nd conference on Winter simulation**

Full text available: pdf(538.55 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes how to take advantage of Internet services and object technology to solve 2D partial differential equations (PDEs) in a distributed manner. This is accomplished by means of a distributed object oriented continuous simulation language designed by our research group, called *OOCSMP,* and a Java (and C++) generating compiler for this language (called *C-OOL*). We also describe a graphical mesh generator, which produces *OOCSMP* code. The mesh generator may be inv ...

6 Streaming 2: ReMDoR: remote multimedia document retrieval over partial order transport

Phillip T. Conrad, Armando Caro, Paul Amer
October 2001 **Proceedings of the ninth ACM international conference on Multimedia**

Full text available: pdf(1.41 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents results from performance experiments that demonstrate and quantify performance improvements when a PO/R transpor5t service is used instead of an ordered/reliable service (O/R e.g., TCP) or an unordered/unreliable service (e.g. UDP). We first describe the *Remote Multimedia Document Retrieval system (ReMDoR)*, an experimental application developed by the authors to evaluate the performance of remote document retrieval over a variety of transport protocols. We then provide ...

Keywords: multimedia, partial order, transport protocols

7 Dynamic points-to sets: a comparison with static analyses and potential applications in program understanding and optimization

Markus Mock, Manuvir Das, Craig Chambers, Susan J. Eggers
June 2001 **Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering**

Full text available: pdf(106.92 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we compare the behavior of pointers in C programs, as approximated by static pointer analysis algorithms, with the actual behavior of pointers when these programs are run. In order to perform this comparison, we have implemented several well known pointer analysis algorithms, and we have built an instrumentation infrastructure for tracking pointer values during program execution.Our experiments show that for a number of programs from the Spec95 and Spec2000 benchma ...

Keywords: alias analysis, calpa, dynamic analysis, points-to analysis, program instrumentation, program optimization, program understanding

8 Coping with type casts in C

Michael Siff, Satish Chandra, Thomas Ball, Krishna Kunchithapadam, Thomas Reps
October 1999 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 7th**

European software engineering conference held jointly with the 7th ACM SIGSOFT international symposium on Foundations of software engineering, Volume 24 Issue 6

Full text available: pdf(1.32 MB) Additional Information: full citation, abstract, references, citings, index terms

The use of type casts is pervasive in C. Although casts provide great flexibility in writing programs, their use obscures the meaning of programs, and can present obstacles during maintenance. Casts involving pointers to structures (C structs) are particularly problematic, because by using them, a programmer can interpret any memory region to be of any desired type, thereby compromising C's already weak type system.This paper presents an approach for making sense of such casts, i ...

9 Teaching experimental design in an operating systems class

Allen B. Downey

March 1999 **ACM SIGCSE Bulletin , The proceedings of the thirtieth SIGCSE technical symposium on Computer science education**, Volume 31 Issue 1

Full text available: pdf(588.84 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes an operating systems (OS) class that departs from more common approaches by introducing experimental design explicitly as part of the course material. Instead of implementing operating systems components or modifying existing operating systems, students conduct a series of experiments that measure the performance of system services and try to infer information about their implementation from the results. These experiments reinforce the OS concepts presented in lecture, and a ...

10 The NuMesh: a modular, scalable communications substrate

Steve Ward, Karim Abdalla, Rajeev Dujari, Michael Fetterman, Frank Honoré, Ricardo Jenez, Philippe Laffont, Ken Mackenzie, Chris Metcalf, Milan Minsky, John Nguyen, John Pezaris, Gill Pratt, Russell Tessier

August 1993 **Proceedings of the 7th international conference on Supercomputing**

Full text available: pdf(1.02 MB) Additional Information: full citation, references, index terms

11 The DOWL distributed object-oriented language

Bruno Achauer

September 1993 **Communications of the ACM**, Volume 36 Issue 9

Full text available: pdf(2.55 MB) Additional Information: full citation, references, citings, index terms

Keywords: concurrency, concurrent object-oriented programming

12 The muse object architecture: a new operating system structuring concept

Yasuhiko Yokote, Fumio Teraoka, Atsushi Mitsuzawa, Nobuhisa Fujinami, Mario Tokoro

April 1991 **ACM SIGOPS Operating Systems Review**, Volume 25 Issue 2

Full text available: pdf(1.92 MB) Additional Information: full citation, abstract, citings, index terms

A next generation operating system should accommodate an ultra large-scale, open, self-advancing, and distributed environment. This environment is dynamic and versatile in nature. In it, an unlimited number of objects, ranging from fine to coarse-grained, are emerging, vanishing, evolving, and being replaced; computers of various processing capacities are dynamically connected and disconnected to networks; systems can optimize object execution by automatically detecting the user's and/or program ...

Results 1 - 12 of 12